

**TECHNICAL SUPPORT FOR THE
CENTER FOR ENTERPRISE INTEGRATION**

**DELIVERY ORDER FOR
GCCS DATABASE MIGRATION**

**NPG MAINTENANCE MANUAL
21 NOVEMBER 1995**

TABLE OF CONTENTS

| <u>TITLE</u> | <u>PAGE</u> |
|---|--------------------|
| SECTION 1 - SCOPE | 1 |
| 1.1 IDENTIFICATION | 1 |
| 1.2 SYSTEM OVERVIEW | 1 |
| 1.3 DOCUMENT OVERVIEW | 1 |
| SECTION 2 - REFERENCED DOCUMENTS | 3 |
| 2.1 SPECIFICATIONS | 3 |
| 2.2 OTHER PUBLICATIONS | 3 |
| SECTION 3 - GENERAL DESCRIPTION | 4 |
| 3.1 EXTERNAL INTERFACES | 4 |
| 3.1.1 Gain Momentum | 4 |
| 3.1.2 UNIX Operating System | 4 |
| 3.1.3 Oracle Database | 5 |
| 3.2 NPG APPLICATION | 5 |
| SECTION 4 - SYSTEM DESCRIPTION | 7 |
| 4.1 SESSION DEFAULTS | 7 |
| 4.2 REQUIREMENTS ANALYSIS | 9 |
| 4.2.1 Filler Requirements | 9 |
| 4.2.2 Replacement Requirements | 11 |
| 4.3 NON-UNIT TPFDD BUILD | 13 |
| 4.3.1 Personnel Working File | 13 |
| 4.3.2 TPFDD Build | 16 |
| 4.3.2.1 Build Button | 18 |
| 4.3.2.2 Edit Button | 19 |
| 4.3.2.3 Merge Button | 19 |
| 4.3.2.4 Delete Button | 19 |
| 4.3.2.5 Other Options | 19 |

TABLE OF CONTENTS (CONT'D)

| <u>TITLE</u> | <u>PAGE</u> |
|--|--------------------|
| 4.3.3 TPFDD Edit | 20 |
| 4.3.4 TPFDD Merge | 23 |
| 4.4 COMMON FUNCTIONS | 25 |
| 4.4.1 Function serviceCode | 25 |
| 4.4.2 Function classification | 26 |
| 4.4.3 Function setClassification | 26 |
| 4.4.4 Function classificationColor | 27 |
| 4.4.5 Function validate_update_privileges | 28 |
| 4.4.6 Printed Reports | 28 |
| 4.4.7 Help | 30 |
| 4.4.8 NPG Release Tool | 31 |
| 4.5 NPG, MEPES AND THE GCCS CORE DATABASE TABLES | 32 |
| 4.5.1 NPG Tables Added to the GCCS Core | 33 |
| 4.5.1.1 NPG_OPLAN_NON_UNIT_RQMT_LOC | 34 |
| 4.5.1.2 NPG_OPLAN_NONUNIT_RQMT_PRSL | 34 |
| 4.5.1.3 PRSL_WORK_FILE | 35 |
| 4.5.1.4 PRSL_WORK_FILE_APOE_ROUTING | 35 |
| 4.5.1.5 PRSL_WORK_FILE_APOD_ROUTING | 36 |
| 4.5.2 NPG Data Element Definitions | 37 |
| 4.5.2.1 APOD Source Indicator | 37 |
| 4.5.2.2 First PIN Sequence Number | 37 |
| 4.5.2.3 Oplan Service Scenario Identification | 37 |
| 4.5.2.4 Personnel Movement Percent | 37 |
| 4.5.2.5 Personnel Working File ID | 38 |
| 4.5.2.6 Personnel Working File Name | 38 |
| 4.5.2.7 Time Increment in Days | 38 |
| 4.5.2.8 Time Period Start Day | 38 |
| 4.5.3 MEPES and GCCS Core Tables | 39 |
| 4.6 DATABASE ACCESS | 40 |

TABLE OF CONTENTS (CONT'D)

| <u>TITLE</u> | <u>PAGE</u> |
|--|--------------------|
| SECTION 5 - SOFTWARE PROGRAMMING ENVIRONMENT | 42 |
| 5.1 GAIN MOMENTUM DEVELOPMENT ENVIRONMENT | 42 |
| 5.1.1 Gain Momentum Installation and Administration | 42 |
| 5.1.1.1 Gain Momentum Installation Guide | 42 |
| 5.1.1.2 Gain Momentum Administration Manual | 43 |
| 5.1.2 Using Gain Momentum | 43 |
| 5.1.2.1 Getting Started Manuals | 43 |
| 5.1.2.2 Applications Programming Manuals | 43 |
| 5.1.2.3 Database Applications Manual | 44 |
| 5.2 NPG APPLICATION DEVELOPMENT AND MAINTENANCE | 44 |
| 5.3 NPG SYSTEM INTEGRATION | 46 |
| 5.3.1 NPG Application Segment | 46 |
| 5.3.2 NPGDB Database Segment | 48 |
| 5.3.3 Installing NPG and NPGDB | 50 |
| 5.3.4 NPG and NPGDB Post Installation | 50 |
| SECTION 6 - ACRONYMS AND ABBREVIATIONS | 53 |

LIST OF FIGURES

| | |
|---|----|
| Figure 3-1: NPG External Interfaces | 4 |
| Figure 3-2: NPG Flow of Control | 5 |
| Figure 4-1: NPG Database Relationships | 33 |
| Figure 4-2: Database Access | 40 |
| Figure 5-1: NPG Runtime Directory Structure | 47 |
| Figure 5-2: NPG Segment Directory Structure | 48 |
| Figure 5-3: NPGDB Segment Directory Structure | 49 |

LIST OF TABLES

| <u>TITLE</u> | <u>PAGE</u> |
|--|--------------------|
| Table 4-1: Session Defaults Error Messages | 8 |
| Table 4-2: Filler Requirements Error Messages | 10 |
| Table 4-3: Replacement Requirements Error Messages | 11 |
| Table 4-4: Personnel Working File Error Messages | 13 |
| Table 4-5: TPFDD Build Error Messages | 17 |
| Table 4-6: TPFDD Edit Error Messages | 21 |
| Table 4-7: TPFDD Merge Error Messages | 24 |
| Table 4-8: NPG Release Tool Error Messages | 31 |

SECTION 1 - SCOPE

1.1 IDENTIFICATION

The purpose of the Non-unit Personnel Generator (NPG) Maintenance Manual is to provide the information necessary to enhance and maintain the NPG Computer Software Component (CSC). NPG is a component of the Global Command and Control System (GCCS) Computer Software Configuration Item (CSCI). This manual was developed for the GCCS Database Migration under Contract Number DCA100-94-D-0016.

1.2 SYSTEM OVERVIEW

NPG was developed using Gain Momentum. Gain Momentum is Sybase's multimedia development system. It operates in a multi-user environment and provides the tools necessary to create Oracle database applications, either with the Graphical User Interface (GUI) or the Gain Extension scripting Language (GEL). Gain Momentum is part of the Government's Common Operating Environment (COE).

NPG consists of Requirements Analysis and Time-Phased Force Deployment Data (TPFDD) Generation. Requirements Analysis enables users to determine both Filler and Replacement requirements under specific conditions. TPFDD generation provides users with the tools to build and modify TPFDD files and merge them into OPLANs on the GCCS Core Database.

NPG accesses Medical Planning and Execution System (MEPES) data within the MEPES Database tables, and stores its own data in NPG tables within the GCCS Core Database. NPG requires two segments to be present on the GCCS Database server: the GCCS Database segment and the MEPES Database segment. The GCCS segment contains the Operation Plans (OPLANs) that NPG may need to update, and MEPES contains the Medical Working Files (MWFs) from which NPG gets most of its input data.

1.3 DOCUMENT OVERVIEW

This manual provides the maintenance programmer with a description of the NPG architecture and design, the NPG development environment, and NPG systems integration. It is intended to bridge the gap between a function's operation and the code. To perform the maintenance function, the user must be familiar with the Gain Momentum development environment, Gain Extension Language (GEL), Structured Query Language (SQL), Oracle Relational Database, and UNIX Operating System.

Section 1 provides a system overview.

Section 2 identifies documents referenced.

Section 3 contains a general description of the NPG capabilities and illustrates the system architectural dependencies.

Section 4 contains a detailed description of the NPG application.

Section 5 describes the development environment.

Section 6 lists the abbreviations and acronyms used within this document.

SECTION 2 - REFERENCED DOCUMENTS

The documents listed below were referenced when preparing this manual. Should a conflict occur between these documents and the contents of the NPG Maintenance Manual, the maintenance manual shall take precedence.

2.1 SPECIFICATIONS

Individual Manpower Requirements and Availability System (IMRAS) Preliminary Software Requirements Specification. SRA Corporation, Arlington, VA, October 1992.

User Interface Specifications for the Global Command and Control System (GCCS). Navy Command Control and Ocean Surveillance Center Research, Development, Test & Evaluation Division (NRaD), San Diego, CA, October 1994.

2.2 OTHER PUBLICATIONS

MEPES Core Users Manual. SRA Corporation, Arlington, VA, July 1994.

MEPES Core Technical Manual. SRA Corporation, Arlington, VA, July 1994.

JOPS III Non-unit Personnel Generator (NPG) Users Manual. JDSSC, 13 January 1989.

Gain Momentum Gel Technical Reference Manual, Volumes 1 - 6. Sybase, Inc., 15 June 1994.

Gain Momentum Developer's Guide to Gel, Volumes 1 and 2. Sybase, Inc., 15 June 1994.

Gain Momentum User's Guide. Sybase, Inc., 15 June 1994.

Gain Momentum Developing Database Applications. Sybase, Inc., 21 February 1995.

Gain Momentum System Administration Manual for Solaris. Sybase, Inc., 15 March 1995.

System Services Maintenance Manual. Boeing Information Systems, Inc., 8 September 1995.

SECTION 3 - GENERAL DESCRIPTION

This section provides a high-level description of the external interfaces to NPG, the NPG application, the NPG database, and NPG integration into the GCCS environment. Sections 4 and 5 of this manual contain the details of these items.

3.1 EXTERNAL INTERFACES

NPG has three primary external interfaces: Gain Momentum, the UNIX Operating System, and the Oracle Database. Figure 3-1 shows the relationship between these three components.

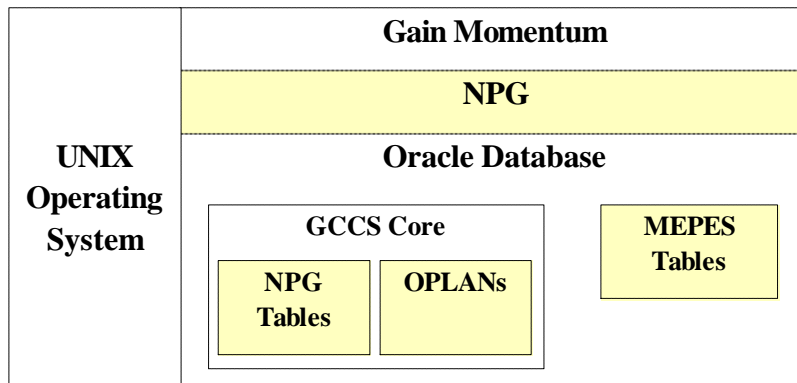


Figure 3-1: NPG External Interfaces.

3.1.1 Gain Momentum

A Gain Momentum runtime is a stand-alone application created by using the Gain Momentum development environment. Each runtime is similar to the Gain Momentum development environment; the main difference is that the runtime includes only the object libraries and data files that are necessary to run the application.

Gain Momentum is included as an external interface because it is not part of the NPG business application. To Gain Momentum, NPG is a runtime; to the user, NPG is a GCCS JOPES application. Both perspectives are correct. The NPG runtime contains the NPG application as well as the NPG boot file, shell scripts, available fonts, etc., that setup the NPG application when it is invoked by the user.

3.1.2 UNIX Operating System

The UNIX operating system provides the environment in which NPG runs. This interface is necessary to create directory and file permissions, supply tools to support the development and maintenance of NPG, configure peripheral devices such as printers, provide an interface to the

network, etc. The scope of this manual does not include a discussion of UNIX systems administration, but describes in Section 5 the UNIX interface as it applies to setting up the Gain Momentum environment, i.e., setting file permissions or establishing NPG accounts.

3.1.3 Oracle Database

The Oracle database consists of the GCCS Core and the MEPES tables. The GCCS Core contains, among other things, the NPG-specific tables and OPLAN data which NPG reads and writes. The MEPES tables contain the MWFs and related data that NPG uses as input for its functions. NPG users have read-only privileges to the MEPES tables, but have read and write privileges to the NPG tables.

3.2 NPG APPLICATION

The NPG application consists of three primary components: Session Defaults, Requirements Analysis, and Non-unit TPFDD Build. Requirements analysis is further divided into the Filler and Replacement requirements functions, and the Non-unit TPFDD build consisting of the TPFDD Build, Personnel Working File (PWF), and Non-unit Personnel TPFDD functions. Users access all components through Session Defaults as illustrated in Figure 3-2.

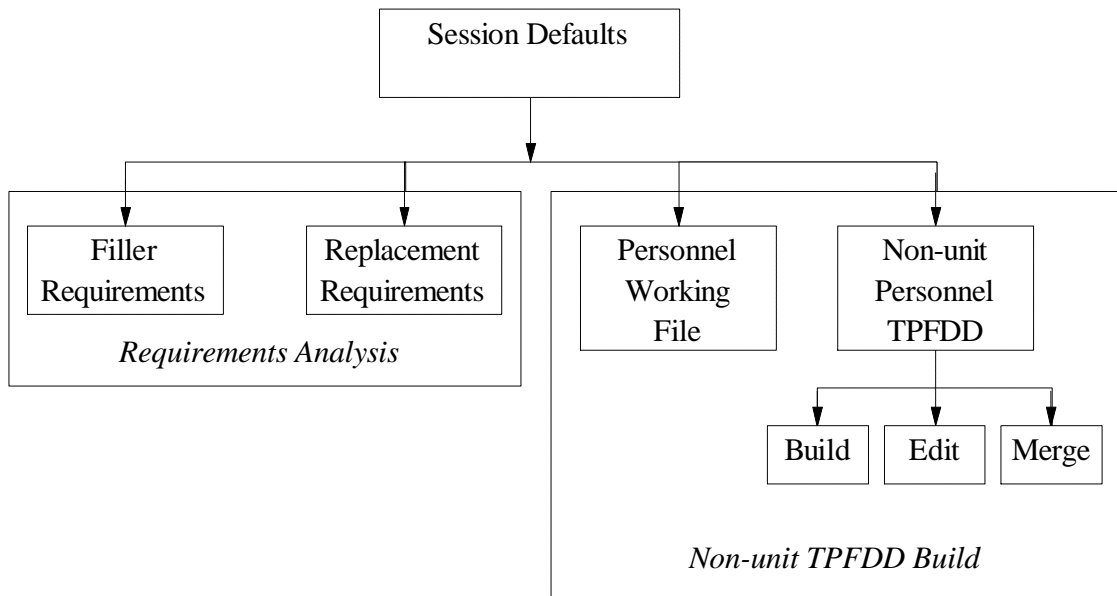


Figure 3-2: NPG Flow of Control.

Session Defaults is the first screen the user sees. It requires that the user specify which Service, OPLAN, and Deployment Operation Commencement Day (C-Day) range to pass to the NPG functions. After selecting the session defaults, the user can choose any of the main functions through the menu bar. In addition to the specified elements, Session Defaults also passes the security classification of the OPLAN to each function, where it is displayed on subsequent screens.

Filler requirements are those personnel of suitable grade and skill initially required to bring a unit to its authorized strength within a stated country. Replacement requirements are those personnel required to replace a stated percentage of losses from all causes of units in theater. The MEPES user determines the percentage of personnel to replace and stores it in the MEPES MWF. Both sets of requirements, filler and replacement, are calculated for a given service, OPLAN, and C-Day range.

The PWF stores information necessary to create Personnel Increment Numbers (PINs). This information includes the Point of Origin, Air Port of Embarkation (APOE), and Air Port of Debarkation (APOD). The combination of the Origin, APOE, and APOD is a routing that designated replacement will follow. Each routing in the PWF contains a planner-specified percentage of replacements. The percentages of all routings added together must total 100%. The PWF also contains a time increment over which the replacements will be aggregated.

The TPFDD Build function is further divided into three additional functions: Build, Edit, and Merge. TPFDD Build combines the routing stored in the PWF with other data in the MWF to create PINs. The accumulation of the PINs is called a TPFDD, and the TPFDD Merge function merges the PINs with any user-selected OPLAN (residing on the GCCS Core Database) to which the user has permissions. Finally, the Edit function allows users to change the values that constitute each PIN.

NPG also has support functions that transcend two or more of the NPG functions. These are listed below. Section 4 of this manual contains a more detailed description of each one.

Print Reports provides preformatted reports for the PWF and TPFDD Build functions.

Print Screen prints the screen that the user is currently on (has focus).

Screen Help contains online documentation describing how to use each screen.

Using NPG is another help function that provides documentation on how to use the NPG application.

SECTION 4 - SYSTEM DESCRIPTION

This section provides a detailed look at each NPG function. The description is divided into six sections:

Purpose of the NPG function.

External interfaces - Inputs and outputs of the function.

Parameters - Input parameters that are passed to the function, much like arguments are passed to subroutines. Mechanically, Gain Momentum passes these parameters by making them attributes of the receiving function. Although they do not appear in the input section of each function, they are still input to the function.

Error detection - A list of errors and their meanings that the NPG function generates.

Algorithms - A review of each functions' process.

Supporting modules - Common functions that are available to some or all of NPG.

4.1 SESSION DEFAULTS

Purpose

Session Defaults is a Gain Momentum application that enables the user to provide default values to use during an NPG session. These defaults, which are propagated to all subsequent screens in NPG, are the OPLAN, service, and start and stop day fields. The Gain Momentum application name for this NPG function is session_defaults.

External Interfaces

User

MEPES Tables: OP_SVC_PAR

Parameters

None

Error Detection

Table 4-1: Session Defaults Error Messages.

| ALERT OR ANNOUNCEMENT | MEANING |
|--|---|
| Alert: You did not successfully log in to the database. | The user must have an Oracle account and must be given permission to execute NPG. |
| An NPG function cannot be chosen because Service, Plan, Start day, or Stop day does not have a value. | NPG requires that the user has values for these fields before continuing. |
| Start day (or Stop day) must be an integer between 0 and 180, inclusive. | C-Day can only be between 0 and 180, inclusive |
| Start day cannot be greater than stop day. | The user must enter a start day that is less than or equal to the stop day. |
| An NPG function cannot be chosen because Plan does not have a value. | The user must select an OPLAN before proceeding to any NPG function. |
| You must choose a service. | NPG requires that the user select a service before selecting an OPLAN. |
| No plans exist for given service. | There are no OPLANS in the database for the service selected. |
| Alert: Operand argument '<entered value>' not a number. Note: Other information may accompany this alert. Alert: This field must be of type integer. Error: Invalid value for DI dataType integer. Note: Other information may accompany this alert. | Either of these messages can occur when the user enters a non-numeric value in a numeric field. |
| Alert: There is a problem locating table op_svc_par in the database. | All Plans shown to the user, for a given Service, are retrieved from table "op_svc_par." If this table does not exist, the user cannot choose a Plan. |

Algorithms

| | | | |
|-------------|-----------|---------|----------------|
| User Input: | OPLAN ID | Output: | OPLAN ID |
| | Service | | Classification |
| | Start Day | | Service |
| | Stop Day | | Start Day |
| | | | Stop Day |

Session Defaults primarily accepts and validates user input for the required defaults, then passes these parameters to subsequent NPG functions. The user cannot proceed from the Session Defaults screen without having valid values for all of these parameters.

Service: The user selects the service (Army, Navy, Air Force, or Marines) via an Option Button Widget. Session Defaults requires that the user select the service before selecting the OPLAN to limit the number of OPLANs that the user has to choose from. This eliminates the possibility of selecting an invalid service-OPLAN combination. Session Defaults converts the service to a service code.

OPLAN ID: Once the user selects the service, Session Defaults queries the database table op_svc_par and returns a list of the OPLANs and their classifications for that service. The user must select one plan from the list.

Start Day and Stop Day: These two fields are related in that the start day \leq stop day, and both fields must be between 0 and 180 inclusive. The user enters these values and Session Defaults validates them as numeric and within the stated range.

Classification: The OPLAN classifications (Unclassified, Confidential, and Secret) were returned with the list of OPLANs.

Session Defaults displays all of the input values to the Session Defaults screen. When the user chooses one of the NPG functions from the menu bar, Session Defaults passes the OPLAN, service code, start and stop days, classification, and classification color to the chosen function.

Supporting Modules

Function serviceCode
Function classification

4.2 REQUIREMENTS ANALYSIS

4.2.1 Filler Requirements

Purpose

Each unit assigned to a specific country has an authorized strength and a current strength. The difference between these two strengths (authorized - current) is called the personnel filler requirement for that unit. Filler personnel are those individuals required to bring a unit up to its authorized strength. The purpose of this function is to calculate the Filler Requirements for any given country. The Gain Momentum application name for this NPG function is filler_requirements.

External Interfaces

User
NPG Views: npg_filler_view

Parameters

OPLAN ID
 OPLAN classification
 Service
 Start Day
 Stop Day

Error Detection*Table 4-2: Filler Requirements Error Messages.*

| ALERT OR ANNOUNCEMENT | MEANING |
|---|--|
| You must enter a country code. | The user tried calculating Filler Requirements before entering a country code. |
| No data is available with given plan, service, country code, and time. | There may not be any requirements for the given country, or for the selected time period, or for the selected service. |
| Country code does not exist. | The user entered an invalid country code on the Filler Requirements screen. |
| Alert: There is a problem locating table geographic_location (or view npg_filler_view) in the database. | The table (or view) does not exist, or the user is not properly logged in. |

Algorithms

| | | | |
|-------------|--------------|---------|----------------------|
| User Input: | Country Code | Output: | OPLAN ID |
| | | | OPLAN Classification |
| | | | Service |
| | | | Start Day |
| | | | Stop Day |
| | | | Country Code |
| | | | Authorized Strength |
| | | | Current Strength |
| | | | Filler |

The user invokes this function from Session Defaults by selecting **NPG, Filler Requirements** from the menu bar. The Filler Requirements function presents the screen and displays the OPLAN ID, Service, and start and stop day range. At this time, the user must enter a Country Code for in-place forces, then click on the Calculate Difference button. When the Filler Requirements function detects the button press it validates the user input and queries the npg_filler_view.

The Filler Requirements function first selects all rows from npg_filler_view for the given OPLAN, service, and start and stop day range. It then sums the authorization (AUTH) and assigned (ASGD) columns. The function subtracts the assigned from the authorized to find the difference between the two and displays all three values on the screen. **Note:** See Section 5 for more information about views.

Supporting Modules

Function serviceCode

4.2.2 Replacement Requirements

Purpose

Replacement Requirements calculates the number of replacement personnel required within a specified Opzone. It calculates for each 10 day interval and for each mission type: either combat and support. This function uses a MEPES MWF as the source of personnel attrition rates. It's Gain Momentum application name is replacement_requirements.

External Interfaces

User

NPG Views: npg_rep_requirements

Parameters

OPLAN ID

OPLAN classification

Service

Start Day

Stop Day

Error Detection

Table 4-3: Replacement Requirements Error Messages.

| ALERT OR ANNOUNCEMENT | MEANING |
|--|--|
| You must choose a Medical Working File and Opzone. | On the Replacement Requirements screen, the user tried to calculate replacement requirements before selecting an MWF and Opzone. |
| You must choose a Medical Working File. | On the Replacement Requirements screen the user tried to select an Opzone before selecting an MWF; the MWF must be selected first. |

| ALERT OR ANNOUNCEMENT | MEANING |
|---|---|
| You must choose an Opzone. | On the Replacement Requirements screen the user tried to calculate replacement requirements without selecting an Opzone. |
| No Medical Working Files found. | There were no MWFs created for the OPLAN and Service selected on the Session Defaults screen. This message appears on the Replacement Requirements screen when the user tries to select an MWF. |
| Alert: There is a problem locating view npg_rep_requirements in the database. | The view does not exist, or the user is not properly logged in. |
| Alert: A SQL error has occurred. | This error could mean many things: the user is not properly logged in; a table or view does not exist; or any other error that can occur when GAIN tries to interface with Oracle through SQL. |

Algorithms

| | | | |
|-------------|------------------|---------|---|
| User Input: | MWF ID Opzone | Output: | Time Period Combat PAR Support PAR Combat Replacement Requirements Support Replacement Requirements |
|-------------|------------------|---------|---|

The user invokes this function from Session Defaults by selecting **NPG, Replacement Requirements** from the menu bar. This function presents the screen and displays the OPLAN ID and Service. At this time, the user must select an MWF and Opzone. The Replacement Requirements function requires that the user select the MWF before selecting the Opzone since not all Opzones are contained in a given MWF. The function detects when the user clicked on the arrow next to the MWF list box and then queries the database to get the list of MWFs. The function displays the list of MWFs in the list box and detects when the user selected one of the choices. Now the user can select an Opzone.

When the replacement requirements function detects that the user clicked on the arrow next to the Opzone list box, it retrieves the list of Opzones pertaining to the selected MWF. When it detects that the user has selected an Opzone from the list and that the user has clicked on the Query Database button, it activates a loop based on the number of 10-day increments between the start and stop day time increment. Within the loop, the function queries the npg_rep_requirements view for the OPLAN, Service, MWF, Opzone, start and stop day range, combat mission, and support mission. From the resulting rows of this query, the function takes the average of the PAR column and the sum of the RQMT column over one 10-day interval. It formats and displays the results, then performs the next iteration of the loop until it has calculated and displayed all 10-day intervals.

Supporting Modules

Function serviceCode

4.3 NON-UNIT TPFDD BUILD

Non-Unit Personnel TPFDD creates movement requirements for non-unit replacement personnel that are based on a selected MWF and PWF. The requirements, collectively known as a TPFDD, are built and deleted via the TPFDD Build function. They can be edited via the TPFDD Edit function and can be merged with a GCCS OPLAN via the TPFDD Merge function. The PWF contains routing data for the requirements and is built using the Personnel Working File function.

4.3.1 Personnel Working FilePurpose

The Personnel Working File function allows the user to create, modify, or delete a PWF. The Gain Momentum application name for this NPG function is pwf.

External Interfaces

MEPES Tables: GEOGRAPHIC_LOCATION

NPG Tables: NPG_PRSL_WORK_FILE
 NPG_PRSL_WORK_FILE_APOE_RTG
 NPG_PRSL_WORK_FILE_APOD_RTG

Parameters

LoginID
 OPLAN
 Service

Error Detection

Table 4-4: Personnel Working File Error Messages.

| ALERT OR ANNOUNCEMENT | MEANING |
|--|--|
| Alert: Problem occurred restoring the previous values. | Could not restore display to its previous value. |
| Alert: First select a Personnel Working File. | Did not select a PWF to print. |
| Alert: SQLErrorMessage() | Login to Oracle failed. |

| ALERT OR ANNOUNCEMENT | MEANING |
|--|---|
| Alert: "Error on commit: <value>", sqlErrorInfo(LoginID) | Could not save changes to the database. |
| Alert: "Error on rollback: <value>", sqlErrorInfo(LoginID) | Could not roll back the database changes. |
| Alert: Maximum number of PWF for service. | No more than nine PWFs per service allowed. |
| Alert: You must enter a value for APOE code. | You cannot continue to the next field until you have entered an APOE. |
| Alert: You must enter a value for Origin code. | You cannot continue to the next field until you have entered an Origin. |
| Alert: % of total must specify a positive integer. | Negative numbers and decimals are not allowed. |
| Alert: Origin code: <value> is invalid. | The origin code you entered does not exist. |
| Alert: APOE code: <value> is invalid. | The APOE code you entered does not exist. |
| Alert: APOD code: <value> is invalid. | The APOD code you entered does not exist. |
| Alert: No PWF record exists for the selected name. | Self explanatory. |
| Alert: You must enter APODs for this PWF. Alert: Must enter APODs for this PWF before exit. | Since you selected time-phased APODs, you must select APODs for this PWF. |
| Alert: You must correct the errors first. | Correct errors on APOE screen before proceeding to the APOD screen. |
| Alert: No C-Days available for the selected PWF. Alert: You must select an available C-Day first. | You must select a time increment. |
| Alert: New Day must be an integer between 0 - 180. Alert: Time increment must be an integer between 1 - 999. Alert: PIN start must be an integer between 1 - 99999 | Self explanatory. |

Algorithm

This NPG function uses the Master-Detail relationship described in Gain Momentum's *Developing Database Applications*, Section *Using SQL Data Manager Functions*. Below is a listing of the Data Managers, their corresponding data items and what they do.

apodDM NPG_PRSL_WORK_FILE_APOD_RTG.PRSL_WORK_FILE_ID

apodDM displays the APOD Code, APOD Name, and percent of total for a given PWF, service, and time period. apodDM makes this information visible only when the user chooses Time Phased as the APOD Source and then selects the Time Phased APODS button.

apoeDM NPG_PRSL_WORK_FILE_APOE_RTG.PRSL_WORK_FILE_ID

apoeDM is used by pwfDM in a Master-Detail relationship to display Origin Code, Origin Name, APOE Code, APOE Name, and percent of Total for a given PWF, service, starting PIN #, and time period.

geoDM GEOGRAPHIC_LOCATION.GLC_CD

geoDM is used by apoeDM primarily to validate the geographic location name and code entered by the user when attempting to add a new PWF.

nameDM NPG_PRSL_WORK_FILE.PRSL_WORK_FILE_ID

nameDM is a picklist that gets a list of all available PWFs.

pickNameDM NPG_PRSL_WORK_FILE.PRSL_WORK_FILE_ID

pickNameDM was created to assist nameDM because Gain Momentum requires every picklist to have two Data Managers (refer to Gain Momentum's *Developing Database Applications*, Section *Creating Picklists*).

pickTimeDM NPG_PRSL_WORK_FILE_APOD_RTG.PRSL_WORK_FILE_ID

pickTimeDM is used by the Time Phased APODS screen to allow the user to choose a Time Period given in C-Days.

pwfDM NPG_PRSL_WORK_FILE.PRSL_WORK_FILE_ID

pwfDM populates the fields PIN Start, Time Increment, and APOD Source (initialized to Opzone) when the user selects a PWF. Then pwfDM, which is the Master in the Master-Detail relationship with apoeDM, calls on apoeDM to populate the Origin Code, Origin Name, APOE Code, APOE Name, and % of Total fields based on the PWF, PIN start, time increment, and APOD Source chosen.

pwf2DM NPG_PRSL_WORK_FILE.PRSL_WORK_FILE_ID

pwf2DM ensures that the new PWF and Service combination being inserted into table npg_prsl_work_file is unique.

timeDM NPG_PRSL_WORK_FILE_APOD_RTG.PRSL_WORK_FILE_ID

timeDM is used by Data Manager apodDM to Delete a C-Day on the Time Phased APODS screen.

Supporting Modules

Function serviceCode
Function classification

4.3.2 TPFDD Build

Purpose

The TPFDD Build function creates movement requirements for non-unit replacement personnel based on the PWF and MWF selected by the user. It displays a list of previously created TPFDDs, identifying each TPFDD by the name of the PWF and MWF used to create it. In addition to creating a new TPFDD, the TPFDD Build function allows the user to delete an existing TPFDD.

TPFDD Build incorporates two other TPFDD-related functions: Merge TPFDD and Edit TPFDD. Users invoke these functions on the TPFDD Build screen after a TPFDD has been created. The Edit TPFDD function allows the user to edit any of the fields in the TPFDD (except PIN number) and commit the changes to the database. The TPFDD merge function allows the user to merge the PINs into an OPLAN on the GCCS Core Database.

In addition, two reports are available via TPFDD Build: the NPG Computations Report and the APODs of ULNs by Opzone Report. The Gain Momentum application named for this NPG function is nutb, which stands for Non-unit TPFDD Build.

External Interfaces

User

GCCS Tables: OPLAN_FORCE_RQMT_LOC

MEPES Tables: MEPES_GEOGRAPHIC_LOCATION

OP_SVC_PAR
OP_SVC_PAR_FORCE_RQMT
OP_SVC_PAR_OPERATION_ZONE
OP_SVC_SCEN_RESULTS

OP_SVC_SCEN_RSLTS_SECTOR_CS
 SVC_SCEN_SITE_RPLAC
 SVC_SCEN_SITE

NPG Tables: NPG_OPLAN_NONUNIT_RQMT_PRSL
 NPG_PRSL_WORK_FILE
 NPG_PRSL_WORK_FILE_APOD_RTG
 NPG_PRSL_WORK_FILE_APOE_RTG

GCCS Views: npg_replacement_rate
 npg_rpl_rqmt
 npg_all_apods_view
 npg_available_tpfdds_view
 npg_rpl_rqmts_by_day_view
 npg_rpl_rqmts_by_opzone_view
 npg_theater_opzones_view

Parameters

OPLAN
 Service
 Start Day
 Stop Day
 Oracle Login ID
 Classification

Error Detection

Table 4-5: TPFDD Build Error Messages.

| ALERT OR ANNOUNCEMENT | MEANING |
|--|---|
| Alert: This TPFDD already exists. | During a TPFDD build, a duplicate record was found. The build will be aborted. |
| Alert: There is no APOD routing effective for day XXX. | During a TPFDD build with time-phased APODs, there was no APOD routing in the selected PWF for the last day in a time increment. The build will be aborted. |
| Alert: Percent must be between 1 and 100. | Data entry out of bounds for the percent field on the Opzone-APOD assignment page. |
| Alert: You must select a Personnel Working File. | No PWF was chosen after pressing the Build button. |
| Alert: You must select a Medical Working File. | No MWF was chosen after pressing the Build button, or after selecting a report. |

| ALERT OR ANNOUNCEMENT | MEANING |
|--|--|
| Error: Unable to delete TPFDD: <oracle error msg>. | Oracle error while attempting to delete a TPFDD after pressing Delete button. |
| Error: Error Creating Report: <exit status> <stdout> <stderr>. | The shell script executed to create a report exited with an error status. The exit status and the contents of stdout and stderr are displayed. |

Algorithm

| | | | |
|-------------|-------------------------|--------------|------------------|
| User Input: | MWF Name | User Output: | Available TPFDDs |
| | PWF Name | | |
| | Opzone-APOD assignments | | |

TPFDD Build receives control from Session Defaults when the user selects **NPG, Non-Unit TPFDD, TPFDD Build** from the menu bar. Upon startup, the TPFDD Build function queries the Npg_Available_TPFDDs_View for all TPFDDs related to the session default OPLAN and Service that are in the database. It displays a list of those TPFDDs on the page NonunitPersonnelTPFDDpage and waits for user input. The user may click on the Build, Edit, Merge or Delete button, or select a menubar item to perform a specific TPFDD Build function.

4.3.2.1 Build Button. This button starts a TPFDD build. TPFDD Build prompts the user to select a PWF and MWF to serve as inputs to the TPFDD build function. The function calculates the number of replacement personnel required over time from personnel loss data in the MWF. It then assigns Origins, APOEs and APODs to the replacement requirements based on routing data that is read from the PWF, or is input from the user. TPFDD Build outputs this data to the Npg_Oplan_Nonunit_Rqmt_Prs1 and Npg_Oplan_Nonunit_Rqmt_Loc tables.

The build process functions in more detail as follows:

TPFDD Build reads data for the selected PWF from Npg_Prsl_Work_File. The build proceeds differently based on the value of Npg_Prsl_Work_File.APOD_Source_Icd.

If APOD_Source_Icd = 'T' (Time-phased APODS), TPFDD Build calculates replacement requirements by querying the Npg_Rpl_Rqmts_By_Day_View. This view calculates daily requirements for each day from the session default start day to the session default stop day. It sums together the requirements for all sectors in all theater Opzones.

TPFDD Build then aggregates the daily requirements into requirements per time period using the time increment in Npg_Prsl_Work_File.Time_Increment_Dy_Cqy. For each aggregate requirement, the function reads the set of APODs and distribution percentages that is effective on the last day of the time period from Npg_Prsl_Work_File_Apod_Rtg. Then it allocates each aggregate requirement to

its APODs according to the distribution percentages specified, to form a set of requirement-APOD pairs.

If APOD_Source_Icd = 'O' (Opzone-APODs), TPFDD Build calculates replacement requirements by querying the Npg_Rpl_Rqmts_By_Opzone_View. This view calculates daily requirements by Opzone for each day from the beginning C-Day to the ending C-Day. It then aggregates the daily requirements into requirements per time period using the time increment in Npg_Prsl_Work_File.Time_Increment_Dy_Cqy.

The user specifies the set of APODs and distribution requirements used for the aggregate requirements for each Opzone. TPFDD Build prompts the user to enter this data by popping up the "OpzoneAPODdialogPage" page to display the APODs of the ULNs assigned to each Opzone. The user can delete or add to the default APODs for each Opzone, and must specify a distribution percentage for each Opzone. TPFDD Build queries the Npg_All_APODs_View for the default, then allocates each aggregate requirement to the APODs specified for its Opzone according to the user-specified distribution percentages. This results in a set of requirement-APOD pairs.

The remainder of the build process functions independently of the APOD source. For the specified PWF, TPFDD Build reads the Origins, APOEs, and distribution percentages from Npg_Prsl_Work_File_Apoe_Rtg, and allocates each requirement-APOD pair among the Origin-APOEs according to the specified distribution percentages.

Finally, TPFDD Build uses each requirement-APOD-Origin-APOE set to create a non-unit personnel movement requirement, or PIN. A PIN is represented as a single row in Npg_Oplan_Nonunit_Rqmt_Prsl and a single row in Npg_Oplan_Nonunit_Rqmt_Loc for each location specified in the requirement. NPG creates three Npg_Oplan_Nonunit_Rqmt_Loc records for each PIN: one for the Origin, one for the APOE, and one for the APOD.

4.3.2.2 Edit Button. This button invokes the TPFDD Edit function for the TPFDD selected in the available TPFDDs list. See Paragraph 4.3.3 for a description of the TPFDD Edit function.

4.3.2.3 Merge Button. This button invokes the TPFDD Merge function for the TPFDD selected in the available TPFDDs list. See Paragraph 4.3.4 for a description of the TPFDD Merge function.

4.3.2.4 Delete Button. This button deletes the TPFDD selected in the available TPFDD list from the Npg_Oplan_Nonunit_Rqmt_Prsl and Npg_Oplan_Nonunit_Rqmt_Loc tables.

4.3.2.5 Other Options. The NPG Computations Report menu selection first prompts the user to select an MWF then invokes the Printed Reports function to print the named report. See Paragraph 4.4.6 for a description of the NPG Computations report.

The APODs of ULNs by Opzone Report menu selection first prompts the user to select an MWF then invokes the Printed Reports function to print the named report. See Paragraph 4.4.6 for a description of the APODs of ULNs by Opzone report.

Supporting Modules

Help

Npg_Common_Functions.classification

Npg_Common_Functions.serviceCode

Printed Reports

4.3.3 TPFDD Edit

Purpose

The TPFDD Edit function enables users to modify a TPFDD by adding, deleting or modifying PINs. The Gain Momentum application name for this NPG function is tpfdd_edit.

External Interfaces

User

NPG Views: npg_pins

NPG Tables: NPG_OPLAN_NONUNIT_RQMT_LOC (NPG Location Table)
 NPG_OPLAN_NONUNIT_RQMT_PRSL (NPG Personnel Table)

Parameters

OPLAN

Service Code

PWF ID

MWF ID

TPFDD name

TPFDD creation date

Classification

Error Detection

Table 4-6: TPFDD Edit Error Messages.

| ALERT OR ANNOUNCEMENT | MEANING |
|---|--|
| Alert: Operand argument '<entered value>' not a number. Note: other information may accompany this alert. Alert: This field must be of type integer. Error: Invalid value for DI dataType integer. Note: other information may accompany this alert | Either of these messages can occur when the user enters a non-numeric value in a numeric field. |
| Alert: The 'op_mvtrqt_id' field is read-only and cannot be changed. The presenter has been reset to the original value. | On the edit TPFDD screen, the user tried to change the PIN. The PIN cannot be edited. |
| Alert: Only digits may be entered for <xxx> where <xxx> = PAX, ALD, EAD, LAD, RDD, or DLY. | This error occurs when the user enters non-numeric characters in any of these fields. |
| Providing Organization Code is invalid. | The user entered an invalid Providing Organization Code. |
| Alert: <xxx> Geographic Location code is invalid, where <xxx> = ORIG, POE, POD, DEST, or INT. | The user entered an invalid Geographic Location Code. |
| Alert: <xxx> Mode is invalid, where <xxx> = POE, POD, DEST, or INT. | The user entered an invalid Mode. |
| Alert: <xxx> Source is invalid, where <xxx> = POE, POD, DEST, or INT. | The user entered an invalid Source. |
| Alert: PINs must be seven characters in length. | This error occurs when the user is inserting a requirement and is self-explanatory. |
| Alert: Second character of PIN must be a valid requirement type code. | This error occurs when the user is inserting a requirement and is self-explanatory. |
| Alert: Third through fifth characters of PIN must be digits. | This error occurs when the user is inserting a requirement and is self-explanatory. |
| Alert: This replacement already exists. | Each replacement requirement must be unique. This error occurs when the user is inserting a requirement. |

Algorithms

| | | | |
|-------------|---------------------------|-----------------------|--------------------------------|
| User Input: | Passengers (PAX) | Output: | Passengers (PAX) |
| | Providing Organization | | Providing Organization |
| | Origination Code | | Origination Code |
| | Port of Embarkation (POE) | | Port of Embarkation (POE) Code |
| | POE Mode | | POE Mode |
| | POE Source | | POE Source |
| | ALD | | ALD |
| | Port of Debarkation (POD) | | Port of Debarkation (POD) Code |
| | POD Mode | | POD Mode |
| | POD Source | | POD Source |
| | Earl Arrival Date (EAD) | | Earl Arrival Date (EAD) |
| | Late Arrival Date (LAD) | | Late Arrival Date (LAD) |
| | Personnel Description | Personnel Description | |
| | Destination Code | Destination Code | |
| | Destination Mode | Destination Mode | |
| | Destination Source | Destination Source | |
| | RDD | RDD | |
| | Int Code | Int Code | |
| | Int Mode | Int Mode | |
| | Int Source | Int Source | |
| | Delay (DLY) | Delay (DLY) | |
| | Project Code | Project Code | |

Program Output: Personnel Increment Number (PIN)

TPFDD Edit begins by displaying, in a Gain Momentum Matrix Presenter, all PINs that match the OPLAN, Service, MWF ID and PWF ID. Users can perform three functions: modify PINs, insert PINs or delete PINs.

If the user modifies any of the PIN data displayed on the screen and clicks on the commit button, TPFDD Edit updates the data in the location table, personnel table, or both. It does this via the Npg_Pins view. (See Paragraph 4.6 for a description of the relationship between the presenter, view, and Oracle tables.)

If the user clicks on the Insert Requirement button on the TPFDD Edit screen, the matrix presenter inserts a blank line for the user to enter data and create a new PIN. When the user clicks the commit button, the TPFDD Edit function inserts the new PIN data into the personnel and location tables.

If the user clicks on the Delete Requirement button, TPFDD Edit presents a confirmation screen. Upon user confirmation, the application deletes the data corresponding to the selected PIN from the personnel and location tables.

Supporting Modules

Function serviceCode
Function classification

4.3.4 TPFDD Merge

Purpose

This function merges a TPFDD into an OPLAN residing on the GCCS database. The Gain Momentum application name for this NPG function is merge_TPFDD.

External Interfaces

User

GCCS Tables: SEND_QUEUE

OPLAN_NONUNIT_RQMT_LOC

OPLAN_NONUNIT_RQMT_PRSL

NPG Tables:

NPG_OPLAN_NONUNIT_RQMT_LOC

NPG_OPLAN_NONUNIT_RQMT_PRSL

NPG Views:

npg_pins

Parameters

Login ID

OPLAN

Service

PWF_id

MWF_id

dest_plan

Error Detection

Table 4-7: TPFDD Merge Error Messages.

| ALERT OR ANNOUNCEMENT | MEANING |
|--|---|
| You must select a TPFDD to merge. | The TPFDD to merge was not selected. |
| You do not have permission to update this plan. | The following must be true for the user to have permission to update a given Plan: 1. The user must have 'UPD' privileges in table user_function_permission. 2. The plan must reside locally (table oplan must have a 'Y' in field op_pln_res_icd and a 0 or 6 in field op_pln_ld_stat_cd for given Plan). 3. The user must be given permission to this plan in table user_oplan_permission. |
| There is a problem locating any of the following tables: oplan, user_oplan_permission, user_function_permission, npg_oplan_nonunit_rqmt_loc, npg_oplan_nonunit_rqmt_prsl, oplan_nonunit_rqmt_loc, oplan_nonunit_rqmt_prsl. | This problem occurs because either the user is not properly logged in to the Oracle database, or these tables don't exist in the database. |
| There is a problem locating view npg_pins in the database. | This problem occurs because either the user is not properly logged in to the Oracle database, or this view doesn't exist in the database. |
| Alert: This TPFDD already exists. | Trying to insert a row into table oplan_nonunit_rqmt_loc (or oplan_nonunit_rqmt_prsl) a row that already exists. |
| Alert: SQL error message. | An SQL error has occurred. |

Algorithms

Input: (See Parameters)

Output:

NRNUBT Transaction

The user invokes the merge_TPFDD function by pressing the Merge button on the TPFDD Build screen. It calls the function validate_update_privileges to determine whether or not the user has update privileges to the OPLAN. Then it checks the Npg_Pins view to determine if any new TPFDDs exist. If they do, merge_TPFDD creates the NRNUBT transaction and inserts it into the Send Queue. Finally, it synchronizes the local GCCS location and personnel tables with the NPG location and personnel tables by copying the changes from NPG to the OPLAN tables. **Note:** For

information on how to create this transaction, consult the *GCCS System Services Maintenance Manual*.

Supporting Modules

Function validate_update_privileges

4.4 COMMON FUNCTIONS

The functions in this category are used by one or more of the NPG functions described above. They perform services for the NPG functions (Session Defaults, Filler Requirements, TPFDD Build, etc.) and are housed in a separate Gain Momentum application.

4.4.1 Function serviceCode

Purpose

This function converts the service name to the service code.

External Interfaces

None.

Parameters

Full name of service.

Error Detection

None.

Algorithms

Input: (See parameters) Output: Single-character service code.

The GCCS database stores the service code as a character field of length one. The code used by database tables is A for Army, F for Air Force, M for Marines, and N for Navy. On Session Defaults, the user chooses the Service by its full name. This function converts the full name to its single-character code.

Supporting Modules

None.

4.4.2 Function classification

Purpose

This function converts the classification code to the classification name.

External Interfaces

GCCS Tables: SCTY_CLSN_CD

Parameters

OPLAN ID
Service Code

Error Detection

None.

Algorithms

Input: Classification Code Output: Classification Name

This function gets the classification code from the OPLAN and calls the function setClassification to expand the code into the classification Name: UNCLASSIFIED, CONFIDENTIAL, or SECRET.

Supporting Modules

None.

4.4.3 Function setClassification

Purpose

This function changes the classification in the classification banners of the calling application.

External Interfaces

None.

Parameters

Classification

Error Detection

None.

Algorithms

Input: (See parameters) Output: Classification Name

Set the banners of the calling application to the new classification.

Supporting Modules

None.

4.4.4 Function classificationColorPurpose

This function changes the classification color in the classification banners of the calling application.

External Interfaces

None.

Parameters

Classification

Error Detection

None.

Algorithms

Input: (See Parameters) Output: Classification color

Set the banners of the calling application to red if the classification is secret, blue if the classification is confidential, or green if the classification is unclassified.

Supporting Modules

None.

4.4.5 Function validate_update_privileges

Purpose

This function confirms that the user has privileges to access an OPLAN.

External Interfaces

GCCS Tables: OPLAN
 USER_FUNCTION_PERMISSION
 USER_OPLAN_PERMISSION

Parameters

OPLAN
Login ID

Error Detection

None.

Algorithms

Input: (See Parameters) Output: Privilege Indication - True or False

Determine whether or not the user has update privileges to the OPLAN by checking table User_Function_Permission for the UPD permission. Next, check that the OPLAN is resident: in table Oplan, column op_pln_res_icd must contain a 'Y' and column op_pln_ld_stat_cd must contain a 0 or 6. Finally, check that the user name in table User_Oplan_Permission has permission to access the OPLAN.

Supporting Modules

None.

4.4.6 Printed Reports

Purpose

The Printed Reports function gathers data for, formats and prints three NPG reports: Personnel Working File Summary Report, NPG Computations Report and APODs of ULNs by Opzone Report.

External Interfaces

MEPES Tables: OP_SVC_SCEN_RSLTS_SECTOR_CS
 SVC_SCEN_SITE
 SVC_SCEN_SITE_PERS_RPLAC

NPG Tables: NPG_PRSL_WORK_FILE
 NPG_PRSL_WORK_FILE_APOD_RTG
 NPG_PRSL_WORK_FILE_APOE_RTG

NPG Views: npg_computations_report_view
 npg_mepes_summary_view
 npg_rpl_requirements
 npg_rpl_rqmts_by_day_view
 npg_replacement_rate

Parameters

APODs of ULNs By Opzone Report: Output File Name
 OPLAN ID
 Service Code
 MWF ID
 MWF Name
 Oracle Userid/Password String

NPG Computations Report: Output File Name
 OPLAN ID
 Service Code
 MWF ID
 MWF Name
 Oracle Userid/Password String

Personnel Working File Summary Report: Output File Name
 Service Code
 PWF ID
 PWF Name
 Oracle Userid/Password String

Error Detection

If errors occur executing shell scripts, UNIX returns the exit status of the shell and all output to stderr and stdout to the invoking Gain application. If any SQL*Plus errors occur during report generation, NPG captures the error messages within the report. The Printed Reports function creates no error messages on its own.

Algorithm

| | | |
|-------------------------|---------|---|
| Input: (See Parameters) | Output: | APODs of ULNs By Opzone Report NPG Computations Report PWF Summary Report |
|-------------------------|---------|---|

The Printed Reports function is implemented as a set of shell scripts. These scripts run Oracle SQL*Plus to generate reports in temporary files, and then run enscrip to print the files. Gain Momentum applications execute the script files via the GEL runCommand function. The reports are sent to the printer identified by the \$PRINTER environment variable. Gain applications use the \$SCRIPTS environment variable to determine the location of the script files.

Supporting Modules

None.

4.4.7 Help

Purpose

The Help function displays pop-up help pages containing help on various NPG windows and on NPG itself. The Gain Momentum application name for this NPG function is help.

External Interfaces

User

Parameters

Help topic: Name of the window on which help was requested, or NPG.

Error Detection

None.

Algorithm

The Help function is implemented as a Gain Momentum application called "help". It inputs the name of a help topic from the calling application, and pops up the corresponding page of help text. Gain Momentum uses hot areas to provide a hypertext capability for moving between pages with different help topics. (In Gain Momentum, hot areas are text markers that can be manipulated like objects.)

Supporting Modules

None.

4.4.8 NPG Release ToolPurpose

The NPG Release Tool is a Gain Momentum application developed to aid in the creation of the NPG deliverable system. It collects all of the applications from a designated directory and loads them into a Gain Momentum library. The tool provides a window into this library for the purpose of building a runtime. The Gain Momentum application name for this NPG function is release_tool.

External Interfaces

User
Gain Momentum
Unix Operating System

Parameters

None

Error Detection

Table 4-8: NPG Release Tool Error Messages.

| ALERT OR ANNOUNCEMENT | MEANING |
|------------------------------|--|
| Application is invalid. | The specified application either does not exist or is inaccessible. |
| Application already exists. | The specified application has already been specified for inclusion in the runtime. |
| No application selected. | No application has been selected for deletion. |

Algorithms

When the user invokes the release tool, the tool presents an initial screen. The user can take any one of the following actions:

Add Application - When the release tool detects that the “Add Application” button was pressed, it verifies that the user-entered application on the “Application” line exists and is accessible. If the application does not exist or is not accessible, the release tool displays an error message. If the application does exist and is accessible, the tool adds the application to the applications list box which displays all of the applications to be included in the runtime.

Delete Application - When the release tool detects that the “Delete Application(s)” button was clicked, it checks the applications list box to see if the user selected any applications to delete. If so, it deletes the application(s) from the list box. If the release tool detects that the delete button was pressed but no applications were selected, it displays an error message.

Export Application to Baseline - When the release tool detects that the “Export to Baseline” button was pressed, it exports each of the applications specified in the list box to the directory specified in the “Baseline (IFs)” field. It exports the applications in Interchange Format (IF). This format is roughly equivalent to source code.

Import to NPG - When the release tool detects that the “Import to NPG” button was pressed, it imports all of the IF files in the directory specified by the “Baseline (IFs)” field into a library named by the value in the “Baseline Gain Library” field. The tool places this library in the directory specified in the “Baseline Gain Library Dir” and checks in all of the imported applications into the library.

Go to Release Lib - When the release tool detects that the “Go To Release Lib” button was pressed, it opens the library which was created by clicking on the “Import to NPG” button, and the user has the regular Gain Momentum functions available.

Supporting Modules

None.

4.5 NPG, MEPES, AND THE GCCS CORE DATABASE TABLES

NPG accesses three sets of database tables: MEPES, GCCS Core, and NPG. MEPES contains the MWFs from which NPG gets much of its input. NPG reads data from these tables but does not write to them. NPG does update GCCS Core tables, however. It reads unit-related data from GSORTS views in the Core Database and merges TPFDDs containing non-unit PINs into OPLANs to which users have permissions. NPG also writes data to the NPG tables, specifically PIN data and data associated with PWFs. Figure 4-1 illustrates the relationship between NPG and the database.

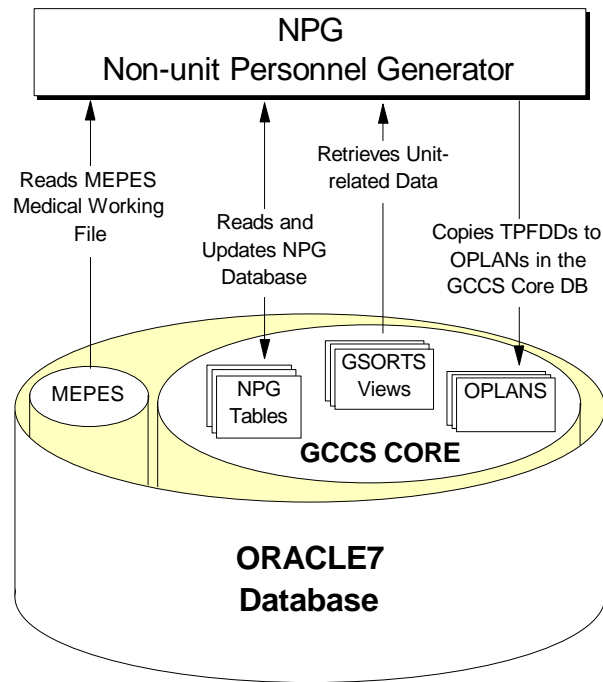


Figure 4-1: NPG Database Relationships.

4.5.1 NPG Tables Added to the GCCS Core

Five tables have been added to the GCCS Core for NPG:

NPG_OPLAN_NONUNIT_RQMT_LOC
 NPG_OPLAN_NONUNIT_RQMT_PRSL
 NPG_PRSL_WORK_FILE
 NPG_PRSL_WORK_FILE_APOE_ROUTING
 NPG_PRSL_WORK_FILE_APOD_ROUTING

The following data items are part of the primary key of all tables containing NPG Non-unit TPFDD Build output. They uniquely identify a PWF - MWF pair that identifies a set of inputs to the Non-unit TPFDD Build.

| | | |
|-------------------|-------|---------|
| PRSL_WORK_FILE_ID | (PWF) | CHAR(1) |
| SVC_CD | | CHAR(1) |
| OPSV_SCN_ID | (MWF) | CHAR(1) |
| OP_PLN_ID | | CHAR(5) |

The remainder of Paragraph 4.5.1 describes the NPG entities, which have corresponding tables in the GCCS core. Each entity is presented by name, definition, data elements, and foreign keys.

4.5.1.1 NPG OPLAN NONUNIT RQMT LOC.**Definition**

This entity describes the routing data for non unit-related personnel movement requirements generated by the Non_unit TPFDD Build. The following list of elements comprise the set of inputs to the execution of the Non_unit TPFDD Build that generated each requirement. Each element is identified with a P if it is a primary key, an F if it is a foreign key, and a PF if it is both.

Elements

| | | |
|----|-----------------------------|-----------|
| PF | PRSL_WORK_FILE_ID | CHAR(1) |
| PF | SVC_CD | CHAR(1) |
| PF | OPSV_SCN_ID | CHAR(1) |
| PF | OP_PLN_ID | CHAR(5) |
| PF | OP_MVTRQT_ID | CHAR(7) |
| P | OP_RTG_TRNPN_LOC_CD | CHAR(1) |
| F | OP_RTG_GLC_CD | CHAR(4) |
| | OPNRQ_TY_CD | CHAR(1) |
| | OP_RTG_TRNPN_MD_CD | CHAR(1) |
| | OP_RTG_TRNPN_SRC_CD | CHAR(1) |
| | OP_RTG_INTLOC_STOP_RSN_CD | CHAR(1) |
| | OP_RTG_NONUNIT_DELY_DAY_CQY | NUMBER(3) |

FOREIGN KEY: (PRSL_WORK_FILE_ID, SVC_CD, OPSV_SCN_ID, OP_PLN_ID, OP_MVTRQT_ID identifies NPG_NONUNIT_RQMT_PRSL
Delete of NPG_NONUNIT_RQMT_PRSL cascades)

FOREIGN KEY: (OP_RTG_GLC_CD identifies GEOGRAPHIC_LOCATION)

4.5.1.2 NPG OPLAN NONUNIT RQMT PRSL.**Definition**

This entity describes nonunit-related personnel movement requirements generated by a Non_Unit TPFDD Build. The following list of elements comprise the set of inputs to the execution of the Non_unit TPFDD Build that generated each requirement.

Elements

| | | |
|----|---------------------------|---------|
| PF | PRSL_WORK_FILE_ID | CHAR(1) |
| PF | SVC_CD | CHAR(1) |
| P | OPSV_SCN_ID | CHAR(1) |
| P | OP_PLN_ID | CHAR(5) |
| P | OP_MVTRQT_ID | CHAR(7) |
| | OPNRQ_PRSL_PRV_ORG_SRC_CD | CHAR(1) |

| | |
|-----------------------------|-----------|
| OP_MVTRQT_PRJ_CD | CHAR(3) |
| OP_MVMTRQT_POE_ALD_CQY | NUMBER(3) |
| OP_MVMTRQT_POD_EAD_CQY | NUMBER(3) |
| OP_MVMTRQT_DEST_RDD_CQY | NUMBER(3) |
| OP_MVMTRQT_POD_LAD_CQY | NUMBER(3) |
| OP_MVMTRQT_POE_EDD_CQY | NUMBER(3) |
| OP_MVMTRQT_POD_FAD_CQY | NUMBER(3) |
| OPNRQ_NORGNC_TRNPN_PRSL_CQY | NUMBER(5) |
| OP_MVTRQT_DSCN_TX | CHAR(31) |
| OPNRQ_CREATE_DT | DATE |
| OPNRQ_CHNG_DT | DATE |
| OPNRQ_RSV_BSLN_TX | CHAR(30) |
| OP_MVTRQT_LOG_ERR_CD | CHAR(1) |

FOREIGN KEY: (PRSL_WORK_FILE_ID, SVC_CD identifies PRSL_WORK_FILE
Delete of PRSL_WORK_FILE cascades)

4.5.1.3 NPG PRSL WORK FILE.

Definition

This entity identifies sets of user_defined parameters for executing Non-unit TPFDD Build. These data sets are called PWFs.

Elements

| | | |
|----|-----------------------|-----------|
| P | PRSL_WORK_FILE_ID | CHAR(1) |
| PF | SVC_CD | CHAR(1) |
| | PRSL_WORK_FILE_NM | CHAR(15) |
| | TIME_INCREMENT_DY_CQY | NUMBER(3) |
| | FIRST_PIN_SEQ_NBR | NUMBER(6) |
| | APOD_SOURCE_ICD | CHAR(1) |

FOREIGN KEY: (SVC_CD identifies SERVICE)

4.5.1.4 NPG PRSL WORK FILE APOE ROUTING.

Definition

This entity contains user_defined Origin-to-APOE routing distribution schemes for individual PWFs. Allowable Origin-APOE combinations are listed with the percentage of total personnel allocated to each pair.

Elements

| | | |
|----|-------------------|-----------|
| PF | PRSL_WORK_FILE_ID | CHAR(1) |
| PF | SVC_CD | CHAR(1) |
| PF | ORIGIN_GLC_CD | CHAR(4) |
| | ORIGIN_GLC_NM | CHAR(17) |
| PF | APOE_GLC_CD | CHAR(4) |
| | APOE_GLC_NM | CHAR(17) |
| | PRSL_MVMT_PCT | NUMBER(3) |

FOREIGN KEY: (APOE_GLC_CD identifies GEOGRAPHIC_LOCATION)

FOREIGN KEY: (ORIGIN_GLC_CD identifies GEOGRAPHIC_LOCATION)

FOREIGN KEY: (PRSL_WORK_FILE_ID, SVC_CD identifies PRSL_WORK_FILE
Delete of PRSL_WORK_FILE cascades)

4.5.1.5 NPG PRSL WORK FILE APOD ROUTING.Definition

This entity contains user-defined, time-phased APOD routing distribution schemes for individual PWFs. APODs are listed along with the percentage of total personnel allocated to them. The start day of the time period in which each APOD-percentage pair is effective is given in each record. The end day of the time period is defined as the day before the next time period starts, if one exists; otherwise, it is the end of the Session Defaults time period.

Elements

| | | |
|----|----------------------------|-----------|
| PF | PRSL_WORK_FILE_ID | CHAR(1) |
| PF | SVC_CD | CHAR(1) |
| P | TIME_PERIOD_START_RLDY_CQY | NUMBER(3) |
| PF | APOD_GLC_CD | CHAR(4) |
| | APOD_GLC_NM | CHAR(17) |
| | PRSL_MVMT_PCT | NUMBER(3) |

FOREIGN KEY: (PRSL_WORK_FILE_ID, SVC_CD identifies PRSL_WORK_FILE
Delete of PRSL_WORK_FILE cascades)

FOREIGN KEY: (APOD_GLC_CD identifies GEOGRAPHIC_LOCATION)

4.5.2 NPG Data Element Definitions

4.5.2.1 APOD Source Indicator.

Mnemonic: APOD_SOURCE_ICD

Long Name: APOD Source Indicator

Data Element Definition: Indicates the source of APOD routing for a PWF.

Values: O = Opzone-APODs
 T = Time-phased APODs

4.5.2.2 First PIN Sequence Number.

Mnemonic: FIRST_PIN_SEQ_NBR

Long Name: First PIN Sequence Number

Data Element Definition: The user-specified sequence number portion of the first PIN to be generated by the NPG. The PIN itself is formed by prefixing the sequence number with the using organization code and the type of movement code. Successive PINS generated by NPG contain sequence numbers one greater than the previous PIN.

Min Value: 1

Max Value: 99999

4.5.2.3 Oplan Service Scenario Identification.

Mnemonic: OPSV_SCN_ID

Long Name: Oplan Service Scenario Identification

Data Element Definition: The Id number of a service MWF, assigned by Mepes, that together with a service code and Oplan Id, uniquely identify an MWF.

Min Value: 1

Max Value: 6

4.5.2.4 Personnel Movement Percent.

Mnemonic: PRSL_MVMT_PCT

Long Name: Personnel Movement Percent

Data Element Definition: The percentage of personnel being routed through the specified port(s).

Min Value: 0

Max Value: 100

4.5.2.5 Personnel Working File ID.

Mnemonic: PRSL_WORK_FILE_ID

Long Name: Personnel Working File ID

Data Element Definition: A software-generated number that, together with a Service Code, uniquely identifies a PWF.

Min value: 1

Max Value: 9

4.5.2.6 Personnel Working File Name.

Mnemonic: PRSL_WORK_FILE_NM

Long Name: Personnel Working File Name

Data Element Definition: The user_specified name for a PWF.

4.5.2.7 Time Increment in Days.

Mnemonic: TIME_INCREMENT_DY_CQY

Long Name: Time Increment in Days

Data Element Definition: The number of days over which replacement personnel requirements are aggregated. Movement requirements for nonunit-related personnel are generated by NPG with EADs and LADs that are a function of this number, the service, and the start and end day session defaults.

Min Value: 1

Max Value: 180

4.5.2.8 Time Period Start Day.

Mnemonic: TIME_PERIOD_START_RLDY_CQY

Long Name: Time Period Start Day

Data Element Definition: The day, relative to C_Day, when an APOD routing distribution becomes effective.

Min Value: 0

Max Value: 180

4.5.3 MEPES and GCCS Core Tables and Views

NPG accesses the following lists of tables and views:

MEPES Tables (Read Only)

MEPES_GEOGRAPHIC_LOCATION
OP_SVC_PAR
OP_SVC_PAR_FORCE_RQMT
OP_SVC_PAR_OPERATION_ZONE
OP_SVC_PAR_OPZONE_SECTOR
OP_SVC_SCEN_RSLTS
OP_SVC_SCEN_RSLTS_SECTOR_CS
SVC_SCENARIO
SVC_SCEN_SITE
SVC_SCEN_SITE_OPERATION_ZONE
SVC_SCEN_SITE_PERS_RPLAC

GCCS Tables (Read and Write)

OPLAN_NONUNIT_RQMT_LOC
OPLAN_NONUNIT_RQMT_PRSL

GCCS Tables (Read Only)

SMDB_GEOGRAPHIC_LOCATION
OPLAN_ACCESS
OPLAN_FORCE_REQUIREMENT
OPLAN_FORCE_RQMT_LOC
UNIT_PERS_STRENGTH

NPG Views

npg_replacement_rate
npg_rpl_rqmt
npg_rep_requirements
npg_all_apods_view
npg_available_tpfdds_view
npg_rpl_rqmts_by_day_view
npg_rpl_rqmts_by_opzone_view
npg_theater_opzones_view
npg_filler_view

4.6 DATABASE ACCESS

NPG uses a Gain Momentum matrix presenter, Gain Momentum data manager, and an Oracle view to access tables in the Oracle database. The presenter and data manager are connected to the view. Together these objects display data derived from Oracle tables in the database. Whenever data in the tables is modified, the Oracle view is automatically updated. To understand the relationship between the matrix presenter, the data manager, and the table views, we will use one of the NPG functions as an example: TPFDD Edit. Refer to Figure 4-1 throughout the discussion of this example.

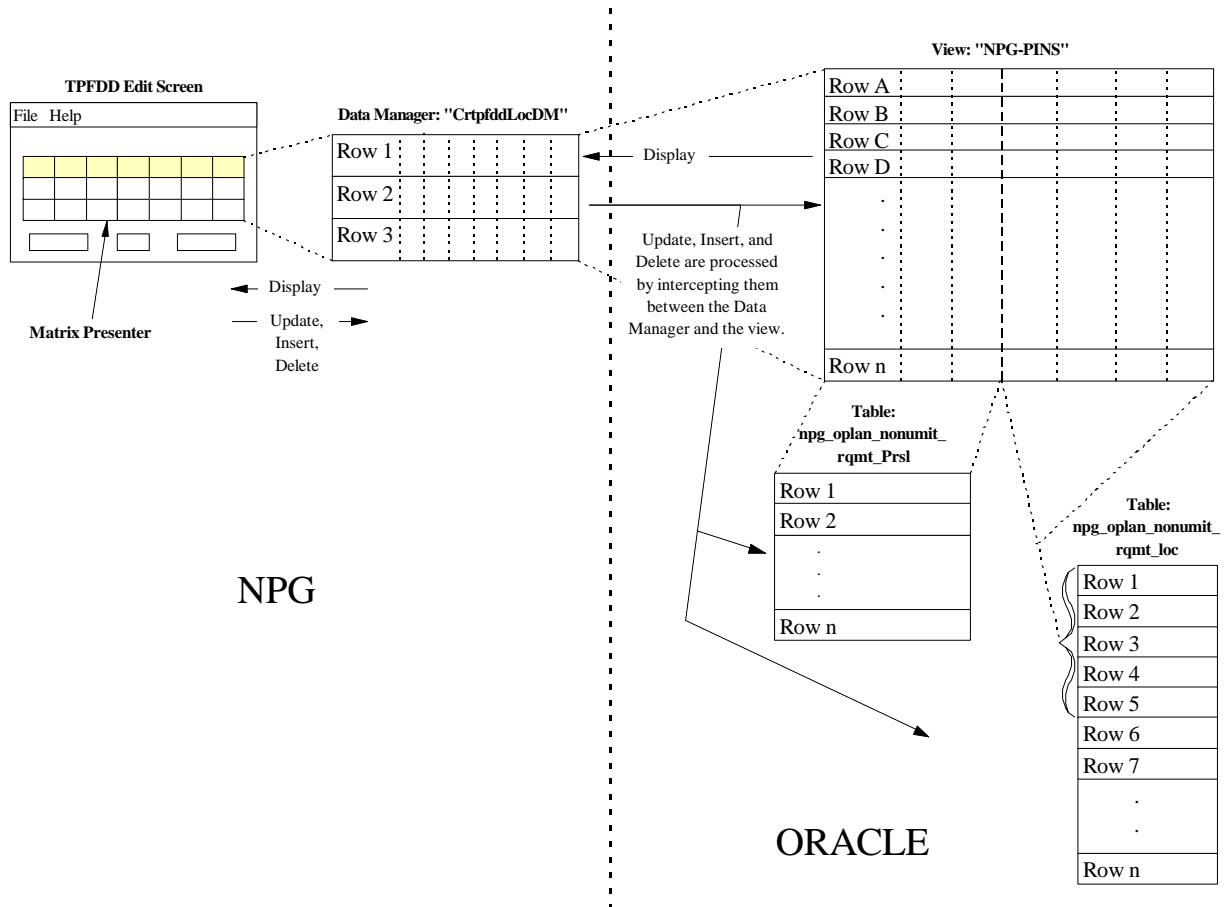


Figure 4-2: Database Access.

In the TPFDD Edit function, the user sees the data via the matrix presenter. This table-like graphic object appears on screen and contains text edit fields, or cells, that enable the user to view, modify or delete the displayed data from the database. The cells are arranged in columns that correspond to the columns of the Npg_Pins view. Additionally, each row in the presenter corresponds to a subset of the rows in the Npg_Pins view. The presenter does not access the database directly, however. Instead, it uses a Gain Momentum data manager.

The Gain Momentum data manager in the TPFDD Edit function is called crtpfddLocDM. It simplifies Oracle database access by providing operations that eliminate the need to use embedded

SQL. CrtpfddLocDM sits between the presenter and the Oracle view, associating the rows and columns of the presenter with the rows and columns of the Npg_Pins view. Based on the user's selection criteria (OPLAN, service, etc.) in effect at any given time, the data in the data manager is a subset of the rows in the Npg_Pins view.

In the TPFDD Edit function, The Npg_Pins view is a view of data that comes from the Oracle tables Npg_Oplan_Nonunit_Rqmt_Prsl (the personnel table) and Npg_Oplan_Nonunit_Rqmt_Loc (the location table). Each row in this view is comprised of exactly one row from the personnel table and zero through five rows from the location table. When records are added to, deleted from or updated in either of these tables, Oracle automatically updates the view.

The data manager ensures that all rows in the Npg_Pins view meeting the user's selection criteria are displayed in the presenter. A change in the Npg_Pins view prompts the data manager to update the presenter. Similarly, a change in the presenter prompts the data manager to update the Npg_Pins view. However, Oracle-built table views cannot be modified. Therefore, the events which trigger the data manager to attempt an update to the Npg_Pins view must be intercepted and applied directly to the personnel and location tables. The events are update, insert, and delete, and Gain Momentum provides "hooks" called preUpdate, preInsert, and preDelete to do the interception, abort the data manager's attempt to process the event, and update the database directly.

The developer has the choice of using either Gain Momentum data managers or Oracle views to access Oracle tables. However, the data managers have an important limitation: they can represent only one Oracle table. If you want to access more than one table using data managers, you need a data manager for each table. You can get around this limitation by using Oracle views to access Oracle tables, since views can represent data in more than one table. In this case, the view becomes the one table that the Data Manager accesses.

SECTION 5 - SOFTWARE PROGRAMMING ENVIRONMENT

This section provides a description of the Gain Momentum development environment, the NPG development environment, and NPG systems integration.

Sybase documents the development environment in a number of documents that help Gain Momentum developers plan projects, use tools, write GEL scripts, use data managers and write SQL code to access the database. Paragraph 5.1 describes each set of manuals briefly to suggest a road map for getting information about the development environment.

Paragraph 5.2 discusses the NPG development environment. Although it is more process than automatic function, it establishes a method for controlling software development and for creating software deliveries.

Paragraph 5.3 describes how to integrate NPG into the GCCS system. This includes descriptions of the scripts that the integration personnel use to install and deinstall the NPG database, set file permissions, set database permissions, etc. It also includes a description of the NPG Runtime and NPG database directory structures.

5.1 GAIN MOMENTUM DEVELOPMENT ENVIRONMENT

The Sybase manuals describe all aspects of the development environment: from installation, setup and administration to development and creating independent runtimes. The intent of Paragraph 5.1 is to provide a brief synopsis of each set of manuals so that developers will know where they can find specific information. We have divided the manuals into two broad categories: installation and administration and using Gain Momentum.

5.1.1 Gain Momentum Installation and Administration

Two manuals provide the information necessary to install, configure, run, and administer the Gain Momentum Development Environment: the Installation Guide and the System Administration Manual.

5.1.1.1 Gain Momentum Installation Guide. To install Gain Momentum on SUN equipment using the Solaris operating system, follow the instructions in the Gain Momentum Installation Guide for Solaris. It identifies the system requirements (hardware, software, memory, disk space, licenses, etc.) for installing Gain Momentum and describes how to install the Gain Momentum software on a networked system. It also contains instructions for installing the Software on a stand-alone machine. This guide provides the installation procedures needed to access either Sybase or non-Sybase databases. Other topics in this manual include installing the software on Hewlett Packard, SunOS, Solaris, IRIX or AIX operating systems; setting up the user environment, permissions, environment variables, libraries, etc.; using the installation tool to install the Gain Momentum software options;

and a list of tips for solving problems that may occur when connecting to non-Sybase database systems.

5.1.1.2 Gain Momentum Administration Manual. The Gain Momentum Administration Manual provides the procedures necessary to make Gain Momentum available to the users and to solve problems that may arise. Users of this manual must have root privileges to the machines where Gain Momentum software resides.

This manual provides an overview of the Gain Momentum system architecture and describes the contents of the Gain Momentum software installation. It describes the daemon processes that are necessary for Gain Momentum to run on a network and discusses the relationship between UNIX and the Gain Momentum software. There are three chapters devoted to creating, using, and manipulating libraries. Additional topics include tidying and recovering files, removing a machine from the network, setting up the database toolkit, managing fonts, configuring a server for Gain Momentum, and a list of UNIX Reference Manual Pages for Gain Momentum Scripts.

5.1.2 Using Gain Momentum

5.1.2.1 Getting Started Manuals. The Gain Momentum Project Guide discusses various generic aspects of the software development life cycle: pre-development activities, design, code development, testing, and deployment. The manual also contains specific information that is important but is not explicitly stated in the developer's or user's guides. For example, it states that developers should create runtimes often to test their code as they progress. They should also test their runtimes on different platforms since the runtimes may react differently to the key mapping, color assignments, and other unique features of the platform.

The Gain Momentum Getting Started document is a brief overview of the development environment. It discusses how to start a new application, how to use the application and folder editors, describes Gain Momentum's scripting language, GEL, and how to include SQL code in the scripts to access the Oracle database.

5.1.2.2 Applications Programming Manuals. The Gain Momentum User's Guide explains the mechanical processes and tools for creating applications in the Gain Momentum development environment. These processes and tools include using the folder, application, timeline, audio, bitmap, link, and script editors; using and modifying foreground, background, and reference pages, text, graphics, and objects; setting access control and permissions and checking applications in and out; creating widgets; and creating runtimes. This document will be most useful during the first six weeks of development for the novice Gain Momentum user, after which it will serve as a handy reference for specific questions and problems.

One of the more important documents is the Developer's Guide to GEL. This document is for developers who will use Gain Momentum GEL scripting to create applications. Whereas the Gain Momentum User's Guide describes how to use tools, such as widgets, the developer's guide describes how to program them into an application.

This manual begins with an introduction to GEL scripting. It describes Gain Momentum objects and containment hierarchy, and defines the GEL variables, handlers and functions, class hierarchy and object properties, operators and statements, and scripting style and standards. Following this is a description of the application development environment. It continues with instructions on manipulating objects, creating and manipulating text and text boxes, and creating and manipulating widgets. These widgets include buttons, menu bars, option buttons, viewers, list boxes, entry widgets, and sliders. There are a few sections towards the end of this guide that describes three features not used by NPG: multimedia, hypermedia, and dynamically linking external objects created with C++. This manual also provides some techniques for tuning applications and defining your own key bindings.

Accompanying the Gain Momentum User's Manual and Developer's Guide is the Gain Momentum GEL Technical Reference Manual. This is like most programmer's Technical reference manuals in that it provides definitive information about all aspects of the GEL programming elements: objects, routines, GEL statements, scripting language syntax, fonts, files, graphics and other constructs.

5.1.2.3 Database Applications Manual. The Gain Momentum Developing Database Applications manual describes how to create applications that require querying, reading or writing data to a database. Since NPG accesses the Oracle database, the Oracle version of this manual is crucial. Developer's will not be able to create applications that access Oracle without it. This manual describes how to use data managers and presenters, create picklists, add GEL scripts to data manager applications, and use SQL to access functions and create database applications.

5.2 NPG APPLICATION DEVELOPMENT AND MAINTENANCE

NPG releases are created using the NPG Release Tool. Section 4 of this manual contains a description of the tools' design, but the following paragraphs describe how to use it.

NPG Release Tool Process

Before starting the NPG Release Tool, each developer having applications to include in the new release should do the following:

Check the applications into the Gain Momentum Library unique to each developer.

Export each application in interchange format by:

- Selecting the application to be exported
- Selecting File, Export, Selection from the main menu
- Specifying a directory, such as, 'tmp' for the Export File field, followed by the name of the application appended with .IF, e.g., /tmp/session_defaults.IF.

Once all of the applications have been checked in and exported, the designated NPG integrator signs on to the Gain Momentum server and uses the following procedure to build the runtime:

Log into the Gain Momentum server as user ‘npg’.

Create a new directory corresponding to the current date under /home/npg/releases, e.g., on October 16, 1995, create /home/npg/releases/101695.

Create a directory called IFs and a directory called gain_libs under the newly created release directory. These will hold the interchange format files and the Gain Momentum library for the release.

Copy the IF files for each application to be included in the runtime from the /tmp directory to the IFs directory.

Launch Gain Momentum (remember to source /home/npg/opstest first) and double click on the release_tool application.

Make sure that each application to be included is listed in the list box at the top of the tool. If any are not, add them by typing them on the Application line and clicking on the Add Application button. Use the Gain convention for specifying an application e.g., app “session_defaults” of library “npg”. If there are any applications in the list box that should be deleted, highlight them by selecting them with the mouse then click on the Delete Application button.

Fill in the “Baseline (IFs)” entry with the name of the newly created directory that holds the IF files.

Fill in the “Baseline Gain Library Dir” with the name of the newly created directory that holds the Gain library.

Click on “Import to NPG” and acknowledge any status messages you receive.

Click on “Go To Release Lib” to open up the library.

Select **File, Make Runtime** from the main menu:

- Under “Startup Actions” select application “session_defaults”.
- Under “Startup Action” select “Start With” and then select “DATA-MANAGER”
- Under “Exit Actions”, set “Exit Gain Momentum with last Sample Editor” to “off” and set “Exit Gain Momentum with last Data Manager” to “on”.
- Under “GEL Script Compilation” set “Keep script source in runtime” to “off” and set “Optimize Scripts in runtime” to “off”.
- Click on “File System Settings”

- Under “Target Directory”, specify the name of the newly created directory up to the date.
- Click on “Make Runtime”
- Answer the questions and wait for completion of the runtime.

Select **File, Quit** to exit the library.

Select **File, Quit** to exit the release tool.

When the release has been built, it is ready to test and deliver. Three scripts have been created to partially automate the process of running a Gain Momentum application on the development machine: setrunglein, setrunpath, and gmOps. Every time a new release is built, you must edit the setrunglein script (currently in the /home/npg/tools directory) to point to the newest runtime. The release tool does not overwrite a previous release; it creates a new one. Therefore it is necessary to "tell" the script where to find the release you want to run; otherwise, the script will run a previous release.

To run the release, source ../setrunpath, source ../setrunglein, then type in 'gmOps'.

5.3 NPG SYSTEM INTEGRATION

One of the last steps in developing and delivering software to the customer is to perform system integration. For NPG, that means integrating two segments: NPG and NPGDB. This section provides an overview of the NPG installation procedure to acquaint developers with the various pieces of the installation process.

5.3.1 NPG Application Segment

In Paragraph 5.2 we described how to create an NPG runtime. What that results in is the directory structure illustrated in Figure 5-1.

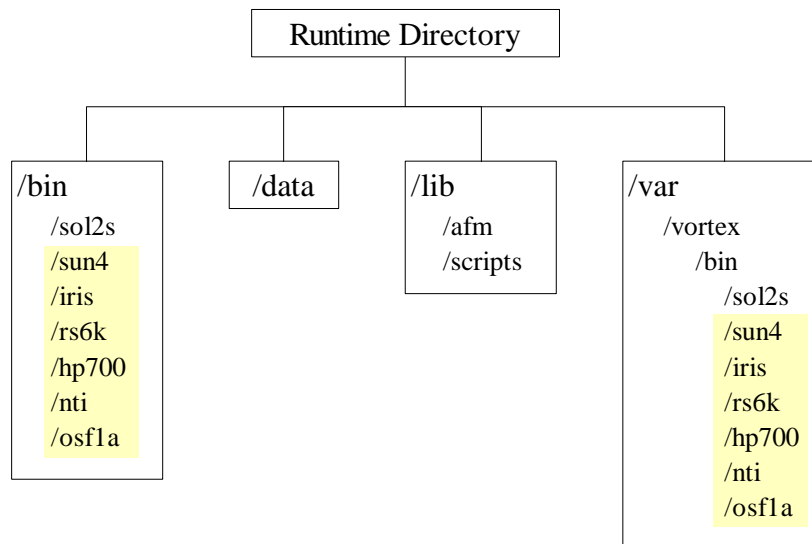


Figure 5-1: NPG Runtime Directory Structure.

The NPG runtime corresponds to executable code and is part of the NPG application segment. As illustrated in Figure 5-1, the runtime contains four primary directories: /bin, /data, /lib, and /var. /bin contains the RDBMS administration shell scripts and the port-specific directories. The only port-specific directory of interest to NPG is sol2s for the Sun Solaris operating system. /data contains the gain boot file and system and runtime libraries. /lib contains the available fonts and GEL scripts that install and startup a runtime. /var contains the port-specific directories for proxy servers, sol2s being the only directory of interest to NPG.

The runtime is a complete unit, and every Gain Momentum runtime contains links to various libraries and files. There may be a way to cut and paste sections of the runtime, but it is not advisable. Even making a one line change to a function within NPG requires building the entire runtime. Although this might seem excessive, it ensures that all relationships, links, mappings, etc. that should exist, will exist. It also ensures that the process for creating runtimes is the same regardless of the scope of the changes. This means that the directory structure in Figure 5-1 will not change unless future releases of Gain Momentum modify it.

Figure 5-1 represents just one directory under the segment directory, NPG. Under NPG, the runtime directory is actually called *progs*. This is illustrated in Figure 5-2 where the entire NPG segment directory is shown.

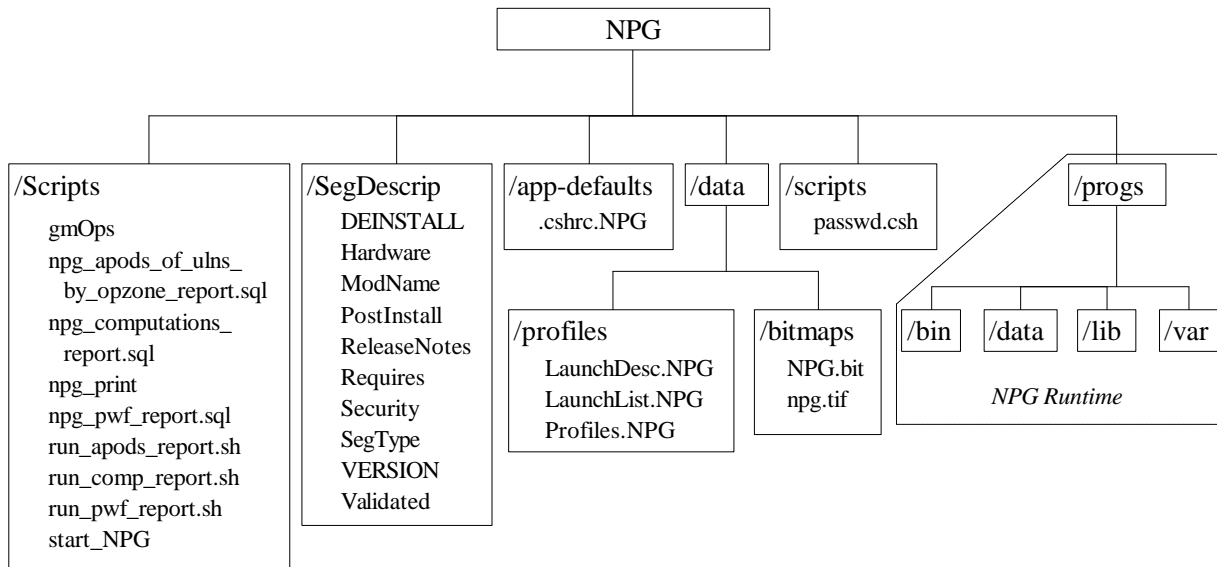


Figure 5-2: NPG Segment Directory Structure.

The Scripts directory contains scripts written to bring up the application (gmOps and start_NPG) and to run the NPG reports (.._report.sql and report.sh scripts). gmOps and start_NPG are run whenever a user invokes NPG, and the report scripts are run within the NPG application when a user opts to print one of the available reports.

The SegDescrip directory contains descriptive information about the application - version number, hardware platform, release notes, segment type, etc. SegDescrip also contains de-installation and post-installation scripts. The de-installation script, DEINSTALL, removes the NPG segment from the system. This script is normally run prior to installing a new version of the NPG runtime. The post-installation script, PostInstall, is run immediately after a new NPG segment has been installed.

The app-defaults and data directories contain environment information that NPG needs to operate on the UNIX server. The data directory also contains the bitmap that NPG displays on startup. Gain Momentum calls it a “splash” screen since it only appears while NPG is being loaded.

The scripts directory contains the script which controls access to the integration process, and the progs directory contains the NPG runtime that we discussed earlier.

5.3.2 NPGDB Database Segment

The second NPG segment is NPGDB, or the NPG database segment. Its directory structure is shown in Figure 5-3.

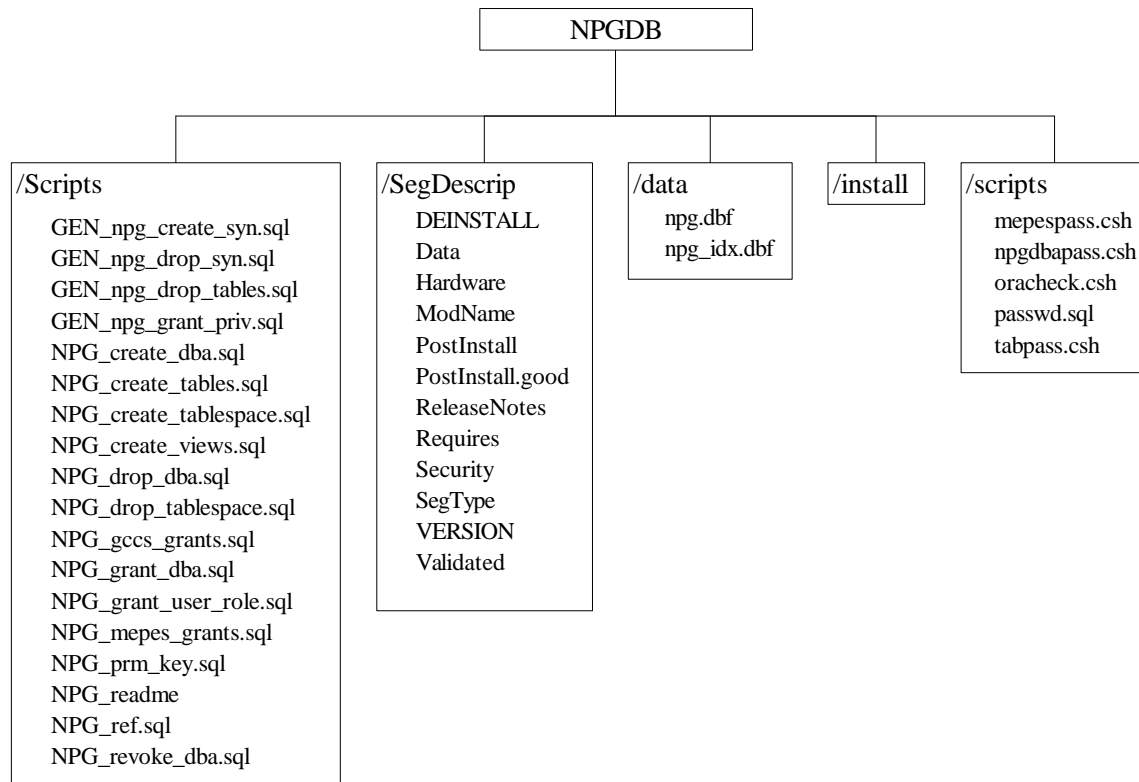


Figure 5-3: NPGDB Segment Directory Structure.

The directory structure of the NPGDB segment is somewhat similar to the directory structure of the NPG segment. The Scripts directory contains scripts written to set up the database application and the SegDescrip directory contains descriptive information about the application, such as, version number, hardware platform, release notes, segment type, etc. It also houses the de-installation and post-installation scripts. In the database segment, however, the post installation script, PostInstall, invokes a number of .sql scripts to assign ownership and permissions, create NPGDBA and NPG users, and create tables, tablespace, and views. The de-installation script, DEINSTALL, removes the public synonyms, tables and tablespace. Both scripts require that the integrator have the proper privileges and know the appropriate passwords to run the post-installation scripts.

NPGDB does not have an app-defaults or progs directory. But like the NPG segment, it does have a data directory. The NPGDB data directory contains actual npg and npg index tablespaces. These files are created during installation when the PostInstall script is run. Also, NPGDB contains a scripts directory that provides security information as does the NPG scripts directory. The information in these scripts is primarily used during post-installation and de-installation.

Once the runtime has been built and all appropriate scripts have been written and placed under the appropriate directories, you must install the NPG and NPGDB segments into the system. We have already described how to configure the runtime into the developers environment, now we will describe how to configure it into a live system.

5.3.3 Installing NPG and NPGDB

The procedure below will install either NPG or NPGDB on a GCCS server. Run the procedure twice if installing both segments, once for each segment.

Login as *sysadmin*. This will automatically start the desktop.

With the mouse, click on SOFTWARE, drag to the JMCIS installer, and release to start the JMCIS installation tool.

Insert the NPG executable segment tape if installing NPG. Otherwise, insert the NPGDB executable segment if installing NPGDB.

Click on READ TOC to read the tapes' table of contents.

Click on the NPG executables to select NPG for installation.

Click on install to start the intallation process.

Click on exit when installation is complete to exit the JCMIS tool.

At this point the software, files, etc. representing NPG and NPGDB are resident on the server, and the the application and the database have been set up. Part of the installation was the automatic process of post-installation and de-installation. Post-installation creates tablespaces, tables and views, grants privileges, sets up passwords, etc. De-installation drops tablespaces, tables and views, removes the NPG administration account, removes passwords, etc. Both NPG and NPGDB have a post-installation and a de-installation script associated with them.

5.3.4 NPG and NPGDB De-Installation and Post-Installation

The post installation scripts partially automate the configurations of the NPG and NPGDB segments into the GCCS system environment. The following lists outline the steps for the post-installation of the NPG segment (NPG DEINSTALL and NPG PostInstall) and for the post-installation of the NPGDB segment (NPGDB DEINSTALL and NPGDB PostInstall).

NPG DEINSTALL

It is likely that prior to configuring the NPG application segment into the GCCS system you will want to de-install the old NPG application configuration if it exists. NPG DEINSTALL does this by performing the following steps and saving the events of the de-installation in \$DEINSTALL_LOG.

Remove the NPG administration account.

Remove the NIS+ password for the NPG administration account.

Remove the auto_home entry for the NPG administration account.

NPG PostInstall

NPG PostInstall configures the NPG application in the GCCS system. It performs the following steps and stores the results in the \$INSTALL_LOG.

Add the NPG administration account.

Set the UNIX password for the NPG administrator in /etc/passwd.

Add the auto_home entry for the NPG account.

Set ownership and permissions for \$INSTALL_DIR and for the runtime directories: /bin, /data, /lib, and /var.

NPGDB DEINSTALL

As with the NPG DEINSTALL, each time you install a new NPGDB segment you may want to de-install the old NPGDB tablespace, tables, and views first. The following list outlines the steps found in the NPGDB DEINSTALL script that removes the old NPGDB configuration from the system and saves the events of the de-installation in \$DEINSTALL_LOG.

Run the scripts that delete the NPGDB tablespace, tables, and views. The scripts are:

- | | | |
|---|-------------------------|--|
| - | NPG_grant_dba.sql | Grant dba privileges to allow the integrator to drop tables, views, and tablespaces. |
| - | GEN_npg_drop_syn.sql | Drop public synonym commands from all tables owned by NPGDBA. |
| - | GEN_npg_drop_tables.sql | Drop NPG tables. |
| - | NPG_drop_tablespace.sql | Drop NPG tablespace. |
| - | NPG_drop_dba.sql | Drop dba privileges to the integrator. |

NPGDB PostInstall

NPGDB PostInstall configures the NPG database in the GCCS system. It performs the following steps and records the events of the installation in \$INSTALL_LOG.

Assign ownership and permissions.

Ensure that Oracle is up and running.

Create Oracle users, NPGDB tables, views, and tablespace by running the following scripts:

- | | | |
|---|---------------------------|---|
| - | NPG_create_tablespace.sql | Create NPG and NPG_IDX tablespace. |
| - | NPG_create_dba.sql | Create the NPGDBA account. |
| - | NPG_mepes_grants.sql | Grant select with grant option to NPGDBA. |

-
- | | | |
|---|------------------------|---|
| - | NPG_gccs_grants.sql | Grant select with grant option to NPGDBA and grant references to NPGDBA. |
| - | NPG_create_tables.sql | Create NPG tables. |
| - | NPG_prm_key.sql | Set primary keys for NPG tables. |
| - | NPG_ref.sql | Set foreign keys for the NPG tables. |
| - | NPG_create_views.sql | Creates the views that NPG uses. |
| - | GEN_npg_create_syn.sql | Generate create public synonym commands for all tables and views owned by NPGDBA. |
| - | GEN_npg_grant_priv.sql | Generate grant commands for public and for the role, NPG_USER. |

Revoke dba privileges from NPGDBA.

There may be other scripts to run that are not part of the automatic installation process. One such script is the NPG_grant_user_role.sql. This script grants the NPG_USER role to existing Oracle users. Look at the readme file or the ReleaseNotes file to determine if there are any scripts that could be run. Generally, scripts mentioned in these files are not required to be run; rather, they are created to automate manual processes that aid the user in some small way.

SECTION 6 - ACRONYMS AND ABBREVIATIONS

| | | |
|--------------|-------|--|
| APOD | | Aerial Port of Debarkation |
| APOE | | Aerial Port of Embarkation |
| ASGD | | Assigned |
| AUTH | | Authorized |
| C-DAY | | Deployment Operation Commencement Day |
| CSC | | Computer Software Component |
| DLY | | Delay |
| EAD | | Earl Arrival Date |
| GCCS | | Global Command and Control System |
| GEL | | Gain Extension Language |
| IF | | Interchange Format |
| IMRAS | | Individual Manpower Requirements and Availability System |
| LAD | | Late Arrival Date |
| MEPES | | Medical Planning and Execution System |
| MWF | | Medical Working File |
| NPG | | Non-unit Personnel Generator |
| NRAD | | Navy Research and Development |
| OPLAN | | Operation Plan |
| PAR | | Population At Risk |
| PAX | | Passengers |
| PIN | | Personnel Increment Number |
| POD | | Port of Debarkation |
| POE | | Port of Embarkation |
| PWF | | Personnel Working File |
| SQL | | Structured Query Language |
| SRA | | Systems Research and Applications Corporation |
| TPFDD | | Time-Phased Force and Deployment Data |